

WHAT IS A CODE REPOSITORY?

- A place to store your code
 - Possibly on your computer, possibly not... but definitely versioned
- A place to *show* your code, and work with others.

WHAT ABOUT ACTUALLY LAYING OUT THE CODE?

- There's not an easy answer for science code - it tends to develop “organically”.
- Often it's best just to split files when they get too big.
- Always keep the novice user (or future you) in mind... Use descriptive names.
- *Think modular!*

WHAT ABOUT PACKAGING CODE?

- Deliver your code in some form that others can install without thinking too hard about where anything goes.
- Makefiles, ruby gems, python packages, etc.
- (Includes sensible versioning!)

WHAT ABOUT PACKAGING PYTHON CODE?



PYTHON PACKAGING

TERMINOLOGY

- “package”: the biggest thing. E.g., *astropy*, *numpy*, *sunpy*. A directory with an “__init__.py”
- “module”: a single “something.py” file - the module is “something”
- “subpackage”: a package within a package
- “source directory/folder”: the directory / folder with all of a codes “stuff”
- “repository” / “repo”: the source directory *in version control*
- “submodule”: a git repo embedded in *another* git repo
- “astropy-helpers”: an example seen in Astropy packages

SAMPLE PACKAGE LAYOUT

README

LICENSE

setup.py

mypackage/__init__.py

mypackage/mymodule.py

mypackage/secondmodule.py

mypackage/subpackage/__init__.py

mypackage/subpackage/anothermodule.py

```
import mypackage
from mypackage import my module
from mypackage import secondmodule
from mypackage import subpackage
from mypackage.subpackage import anothermodule
```


**THE GOAL OF PACKAGING
AND INSTALLING IS
BASICALLY TO MAKE THAT
WORK ANYWHERE**

VERSIONING

- In vogue: “semantic versioning”
- x.y.z (E.g., 0.2.3, 2.7.12, 3.6)
 - change *x* for breaking changes
 - change *y* for non-breaking changes
 - change *z* for bug-fixes
- Anything x.y.z<something else> is a pre-release
 - E.g., 1.2.3beta, 2.1.6rc2
- But don't get too worked up. 0.1 -> 0.2 -> 0.3 is better than nothing.

LICENSING YOUR CODE

- Rule #1: Have a license!
- Rule #2: There is no rule #2.

(see problem sets for more)

GO TO:
[HTTPS://GITHUB.COM/ETEQ/](https://github.com/eteq/python-packaging)
PYTHON-PACKAGING

GET:
PACKAGINGANDREPOS.IPYNB

(MAY WANT TO SKIP TO
PROBLEM 3)